

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ  
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION  
DEPARTMENT OF TELECOMMUNICATIONS

NÁVRH A REALIZACE WEB ROZHRANÍ PRO PBX ASTERISK

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

LUKÁŠ JANDA

BRNO 2013



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH  
TECHNOLOGIÍ**  
**ÚSTAV TELEKOMUNIKACÍ**

**FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION**  
**DEPARTMENT OF TELECOMMUNICATIONS**

# **NÁVRH A REALIZACE WEB ROZHRANÍ PRO PBX ASTERISK**

**DESIGN AND IMPLEMENTATION OF WEB INTERFACE FOR PBX ASTERISK**

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**LUKÁŠ JANDA**

**VEDOUcí PRÁCE**  
SUPERVISOR

**Ing. ONDŘEJ KRAJSA, Ph.D.**

BRNO 2013



VYSOKÉ UČENÍ  
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

Ústav telekomunikací

# Bakalářská práce

bakalářský studijní obor  
Teleinformatika

**Student:** Lukáš Janda

**ID:** 139277

**Ročník:** 3

**Akademický rok:** 2012/2013

## NÁZEV TÉMATU:

**Návrh a realizace web rozhraní pro PBX Asterisk**

## POKYNY PRO VYPRACOVÁNÍ:

Podrobně se seznámte s možnostmi multiplatformního programování pro Internet. Ve Vámi vybraném vývojovém nástroji navrhnete a realizujete webové rozhraní pro správu a konfiguraci softwarové ústředny Asterisk a možností rozšiřování o přídavné moduly. Navrženou aplikaci zabezpečte proti vniknutí neautorizované osoby.

## DOPORUČENÁ LITERATURA:

- [1] KOSEK, Jiří. PHP :Tvorba interaktivních internetových aplikací. Podrobný průvodce /Praha :Grada Publishing,1998. 1. vyd. 490 s. ISBN 8071693731
- [2] NARAMORE, Elizabeth. PHP5, MySQL, Apache: vytváříme webové aplikace / Vyd. 1. Brno : Computer Press, 2006. 813 s. ISBN 80-251-1073-7
- [3] KOFLEER, Michael; ÖGGL, Berndt .PHP 5 a MySQL 5 : průvodce webového programátora, Vyd. 1. Brno : Computer Press, 2007. 607 s., ISBN: 978-80-251-1813-9
- [4] Meggelen, J.V, Smith, J., Madsen, L. Asterisk™ The Future of Telephony. Sevastopol: O'Reilly Media, Inc., 2005. ISBN 0-596-00962-3.

**Termín zadání:** 11.2.2013

**Termín odevzdání:** 5.6.2013

**Vedoucí práce:** Ing. Ondřej Krajša, Ph.D.

**Konzultanti bakalářské práce:**

**prof. Ing. Kamil Vrba, CSc.**

*Předseda oborové rady*

## UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## **ABSTRAKT**

Tato bakalářská práce se zabývá vytvořením webového rozhraní pro správu a konfiguraci softwarové pobočkové ústředny Asterisk. Do aplikace se uživatel dostane po autentizaci. Aplikace je rozdělena na část uživatelskou a část administrační. Pro komunikaci aplikace s ústřednou jsou využívány SQL tabulky.

## **KLÍČOVÁ SLOVA**

Asterisk, PBX, VoIP, Realtime, SIP, IAX2

## **ABSTRACT**

This semestral project is dealing with creating a web interface for managing and configuring software PBX Asterisk. User can use application after autentization. The application is divided into a user section and administration section. For communication with PBX are used SQL tables.

## **KEYWORDS**

Asterisk, PBX, VoIP, Realtime, SIP, IAX2

JANDA, Lukáš *Návrh a realizace web rozhraní pro PBX Asterisk*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 42 s. Vedoucí práce byl Ing. Ondřej Krajsa, Ph.D.

## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Návrh a realizace web rozhraní pro PBX Asterisk“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Ondřeji Krajsovi, Ph.D. za konzultace, trpělivost a ochotu obětovat čas v kteroukoliv denní dobu.

Brno .....

.....

(podpis autora)

# OBSAH

<b>Úvod</b>	<b>9</b>
<b>1 Princip VoIP komunikace</b>	<b>10</b>
1.1 Digitalizace hlasového signálu . . . . .	10
1.2 Přenos hlasu . . . . .	10
1.3 Protokoly . . . . .	10
1.3.1 Protokol H.323 . . . . .	10
1.3.2 Protokol SIP . . . . .	11
1.3.3 Protokol IAX . . . . .	13
1.4 Asterisk . . . . .	13
1.4.1 Asterisk a technologie . . . . .	15
1.4.2 Asterisk RealTime . . . . .	15
1.4.3 Asterisk AMI . . . . .	16
1.4.4 Asterisk AGI . . . . .	17
<b>2 Návrh aplikace</b>	<b>19</b>
2.1 Volba programovacího jazyka . . . . .	19
2.1.1 PHP . . . . .	19
2.1.2 PHP a databáze . . . . .	19
2.2 Propojení aplikace s Asteriskem . . . . .	19
2.2.1 Realtime mód . . . . .	19
2.2.2 „TXT“ mód . . . . .	20
2.3 Členění realtime módu aplikace . . . . .	20
2.3.1 Registrační část . . . . .	21
2.3.2 Uživatelská část . . . . .	21
2.3.3 Administrační část . . . . .	21
2.4 Členění „TXT“ módu aplikace . . . . .	24
2.4.1 New user . . . . .	24
2.4.2 Users management . . . . .	24
2.4.3 Server info . . . . .	24
2.4.4 Dialplan . . . . .	24
2.4.5 Reload Asterisk . . . . .	24
2.5 Zabezpečení aplikace . . . . .	25
2.6 Generování čísla účtu . . . . .	26
2.7 Billing modul . . . . .	27
2.7.1 Uživatelská sekce . . . . .	27
2.7.2 Administrační sekce . . . . .	28

2.8	Příklad procesu žádosti o účet . . . . .	29
2.8.1	Žádost o webový účet . . . . .	29
2.8.2	Žádost o telefonní účet . . . . .	29
2.9	Instalace aplikace . . . . .	31
2.9.1	Požadavky aplikace na sw vybavení serveru . . . . .	31
2.9.2	Import tabulek MySQL databáze . . . . .	31
2.9.3	Konfigurace Asterisku . . . . .	31
2.9.4	Instalace vlastní aplikace . . . . .	33
2.9.5	Nastavení přístupových práv . . . . .	34
<b>3</b>	<b>Závěr</b>	<b>35</b>
	<b>Literatura</b>	<b>36</b>
	<b>Seznam symbolů, veličin a zkratk</b>	<b>37</b>
	<b>Seznam příloh</b>	<b>38</b>
<b>A</b>	<b>Příloha</b>	<b>39</b>
<b>B</b>	<b>Příloha</b>	<b>42</b>



# SEZNAM OBRÁZKŮ

1.1	Výměna zpráv protokolu SIP . . . . .	12
1.2	Komunikace AMI správce událostí . . . . .	16
1.3	Komunikace AMI správce akcí . . . . .	17
1.4	Komunikace AGI s Asteriskem . . . . .	18
2.1	Architektura aplikace . . . . .	20
2.2	Proces přihlašování do aplikace . . . . .	25
2.3	Generování čísla účtu . . . . .	27
2.4	Výpočet ceny hovoru . . . . .	28
A.1	Snímek ze správy uživatelů v TXT módu aplikace . . . . .	39
A.2	Snímek z přidávání uživatelů administrační sekce . . . . .	40
A.3	Snímek Billing modulu . . . . .	41
B.1	Tabulky MySQL databáze . . . . .	42

# ÚVOD

V dnešní době, kdy je kvalitní připojení k internetu dostupné prakticky všude, se čím dál tím častěji pro telefonní komunikaci využívá technologie VoIP, která využívá pro přenos hlasu datových sítí. Nesporná výhoda tohoto řešení přenosu hlasu je cena, platíme prakticky pouze za připojení k internetu. Další výhodou použití VoIP je možnost využít pro komunikaci již existující datovou síť, tudíž nám stačí pouze jedna „kabeláž“ společná pro počítače a telefony.

V mé práci se zabývám návrhem a realizací webového rozhraní, pomocí kterého je možné spravovat a konfigurovat softwarovou ústřednu Asterisk. Má aplikace využívá možnost Asterisku používat tzv. RealTime mód, kdy se propojí ústředna s databází.

# 1 PRINCIP VOIP KOMUNIKACE

## 1.1 Digitalizace hlasového signálu

Dříve se hlas přenášel analogově, VoIP technologie ale analogový signál jako takový nedokáže přenést, je tedy nutné jej digitalizovat. K digitalizaci hlasového signálu slouží A/D převodníky - tzv. kodeky. Kodek není nic jiného než kódovací a dekódovací algoritmus. Ve VoIP se využívají kodeky: G.711, G.722, G.723, G.726, G.729, SPEEX, iLBC...

## 1.2 Přenos hlasu

Řečový signál se po digitalizaci rozdělí na pakety a přenášejí se k cílovému zařízení. Pro přenos signálu se využívá protokol IP ležící na třetí vrstvě OSI modelu a protokol čtvrté vrstvy - UDP. Samotný řečový signál je zakódovaný v protokolu RTP ležícím na páté vrstvě OSI modelu.

Ve VoIP nestačí pouze vysílat pakety obsahující řečový signál, ale je nutné nava-  
zovat a řídit spojení mezi účastníky. Pro tyto účely se využívají protokoly, nejčastěji H.323, SIP, IAX2 (využíván softwarovou ústřednou Asterisk). Jednotlivé protokoly se mezi sebou liší v poskytovaných službách a signalizaci.

## 1.3 Protokoly

### 1.3.1 Protokol H.323

Definuje protokoly pro audio-vizuální komunikace v paketových sítích. V dnešní době je na ústupu. Protokol obsahuje následující komponenty:

#### **Terminál**

Koncový bod umožňující obousměrnou multimediální komunikaci v reálném čase, musí podporovat hlasovou komunikaci. Jedná se o jediný povinný prvek při využití sady protokolů H.323. Může se jednat jak o hardwarové tak o softwarové zařízení.

#### **Brána**

Prvek, který umožňuje propojení H.323 sítí s jinými druhy sítí. Provádí zřízení a ukončení spojení v propojovaných sítích. Není to povinná komponenta. Brána není nutná pouze tehdy, využívají-li obě sítě stejné protokoly.

## Gatekeeper

Převádí adresy do tvaru použitelného pro H.323, má na starosti autentifikaci, autorizaci a řídí přenosovou kapacitu.

## MCU

Jedná se o vícebodové řídicí jednotky - využívá se při komunikaci více než tří terminálů (konferenční hovory). MCU dělíme na dvě části, Multipoint Controller (MC), který se využívá při zřizování spojení, definuje vlastnosti komunikujících terminálů, stará se o navázání/ukončení komunikačních kanálů a Multipoint Processor (MP). MP řeší přepínání videa, zvuku a dat. Na rozdíl od MC je jeho použití nepovinné.

### 1.3.2 Protokol SIP

Jedná se o textový protokol využívaný pro přenos signalizace v IP sítích. Stará se o sestavení, ukončení a modifikaci spojení mezi účastníky.

Protokol SIP využívá protokoly RTP/RTCP a SDP. RTP je používán pro real-time přenos hlasu, SDP popisuje vlastnosti zahajovaného přenosu.

SIP pracuje na principu klient - server, při komunikaci dochází k výměně zpráv žádost a odpověď. Oproti H.323 je výrazně jednodušší, vychází z protokolu http, využívá několik položek podobných položkám SMTP. Na obr. 1.1 je znázorněna výměna zpráv protokolu SIP.

#### Žádosti SIP

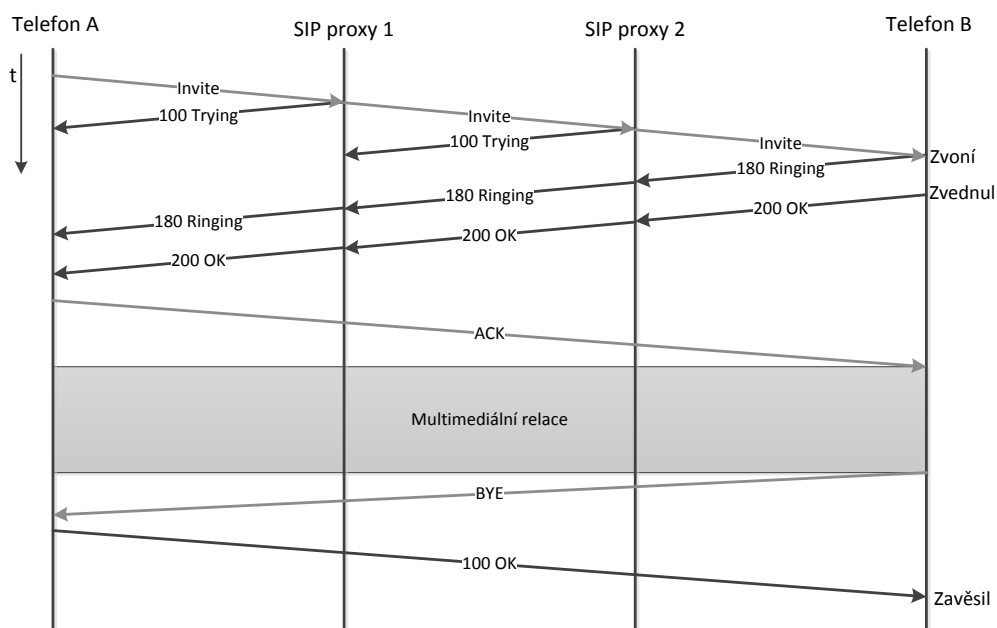
SIP využívá následující metody:

- REGISTER - využívána klienty k registraci jejich současné polohy
- INVITE - zahájení nebo modifikace spojení
- ACK - potvrzení zahájení relace
- CANCEL - zrušení nevyřízené žádosti
- BYE - ukončení relace
- OPTIONS - žádost o informace o možnostech volajícího

## Zprávy SIP

Zprávy jsou odpovědi na žádosti, odpovědi dělíme do následujících šesti skupin

- 1xx - průběh
- 2xx - úspěch
- 3xx - přesměrování
- 4xx - chyba na straně klienta
- 5xx - chyba na straně serveru
- 6xx - fatální chyba



Obr. 1.1: Výměna zpráv protokolu SIP

## Komponenty SIP

### Proxy server

V SIP sítích je zodpovědný za přijímání a přeposílání žádostí od User Agents, nebo od jiných Proxy serverů. Dá se tedy říct, že primární funkcí SIP Proxy serveru je směrování.

## **Redirect server**

Stejně jako Proxy server přijímá zprávy od User Agents, ale na rozdíl od něho zprávy nepřeposílá dál, ale vrací 3xx zprávu s informací (URI), kde se volaný nachází.

## **Registrar server**

Přijímá pouze registrační žádosti od jednotlivých User Agents a aktualizuje podle nich databázi koncových User Agents v rámci domény. V databázi jsou uloženy URI s IP adresami.

## **Adresace**

Adresa v síti SIP je definována pomocí URI, má tvar podobný e-mailové adrese. SIP URI je vždy ve tvaru: sip:uzivatel@domena:port číslo portu bývá standardně nastaveno na 5060.

### **1.3.3 Protokol IAX**

Protokol IAX byl vyvinut firmou Digium jako alternativa k protokolům SIP či H.323 pro přenosy mezi servery Asterisk. V současné době je protokol ve verzi 2 (IAX2). Asterisk byl vědomě navržen pro práci za zařízeními používající NAT. Oproti protokolu SIP probíhá signalizace i samotný hovor na stejném portu (port 4569). Obrovskou výhodou protokolu IAX je trunking - seskupení několika relací do jednoho datového proudu. Trunking umožňuje reprezentovat několik relací pod jednou datagramovou hlavičkou, což napomáhá snížit zpoždění a snižuje požadavky na šířku pásma.

IAX umožňuje vícero možností ověření: plain text, hashování MD5 a RSA výměnu klíčů.

## **1.4 Asterisk**

Asterisk je open source řešení pobočkové ústředny, které je šířeno pod licencí GPL. Pomocí Asterisku můžeme z téměř libovolného počítače (výkonem úměrného plánovanému rozsahu využití) využívajícího platformu Unix (dostupná i pro OS Windows) vytvořit pobočkovou ústřednu, avšak oproti klasické pobočkové ústředně, s mnohem většími možnostmi. Asterisk mimo použití jako pobočkové ústředny umožňuje využití i jako brána mezi různými protokoly používanými při VoIP komunikaci (SIP, IAX2, H.323, Skinny...), využít jej pro Voicemail služby či jako server IVR. Asterisk je navržen tak, aby vytvořil rozhraní telefonnímu hardwaru, softwaru a libovolné telefonní aplikaci.

Asterisk může být mimo jiné použit v těchto aplikacích:[6]

- Různorodá VoIP gateway (MGCP, SIP, IAX, H.323)
- Pobočková ústředna (PBX)
- Voicemail služby s adresářem
- Interaktivní hlasový průvodce (IVR) server
- Softwarová ústředna (Softswitch)
- Konferenční server
- Packet voice server
- Šifrování telefonních nebo faxových volání
- Překlad čísel
- Aplikace Calling card
- Prediktivní volič (Predictive dialer)
- Řazení volání do front se vzdáleným zprostředkovatelem
- Vzdálené „kanceláře“ pro existující PBX

Asterisk podporuje následující kodeky:[4]

- AMR Codec
- BroadVoice Codec
- DoD CELP
- GIPS Family
- GSM
- iLBC
- G.711
- G.722
- G.722.1
- G.722.1C
- G.722.2
- G.723.1
- G.726
- G.728
- ITU G.729
- LPC10
- Speex

### 1.4.1 Asterisk a technologie

Asterisk podporuje 3 základní druhy rozhraní.

#### DAHDI

DAHDI, dříve známý pod jménem Zaptel, je pseudo TDM rozhraní, které pro zpracování TDM využívá namísto specializovaného hardwaru výpočetní výkon CPU. Nespornou výhodou oproti hardwarovému řešení je nižší cena a vyšší flexibilita. Dá se tedy říci, že DAHDI slouží pro propojení Asterisku s klasickou analogovou telefonní sítí.

#### Non-DAHDI

Tato rozhraní připojují Asterisk ke klasické telefonní sítí, avšak bez podpory pseudo TDM.[6]

#### Packet voice

Na rozdíl od DAHDI a Non-DAHDI se stará o připojení k paketové síti. Pro komunikaci v síti využívá svůj vlastní protokol IAX2, podporuje však další běžně používané packet voice protokoly:

- SIP
- H.323
- MGCP
- VoFR.

### 1.4.2 Asterisk RealTime

Asterisk běžně uchovává konfigurační soubory na serveru, na kterém je nainstalován, architektura RealTime umožňuje uchovávat vybrané konfigurační soubory včetně jejich nastavení v databázi. Rozlišujeme dva typy RealTime: dynamic a static.

#### Static realtime

Statická verze je v podstatě identická s tradiční metodou používání konfiguračních souborů, rozdíl je pouze v umístění konfiguračních souborů (konfigurační soubory Asterisku/tabulky v databázi).



## Dynamic realtime

Dynamická verze se používá tam, kde je potřeba aby Asterisk „zaregistroval“ změnu konfigurace ihned po provedení změny v příslušné tabulce databáze. To znamená, že využití dynamic realtime se přímo vybízí pro uchovávání nastavení konfiguračních souborů typu SIP, IAX, Extensions...

### 1.4.3 Asterisk AMI

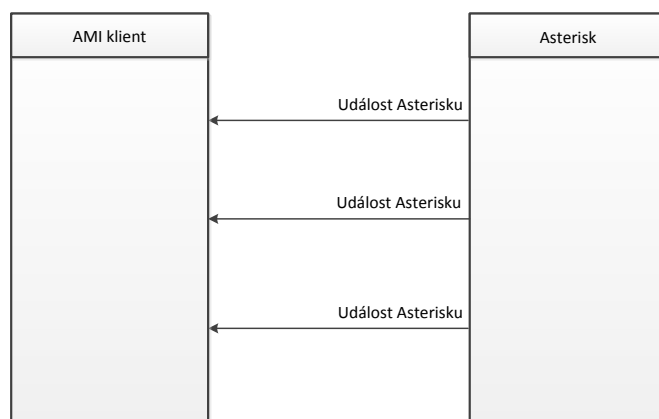
Asterisk Manager Interface umožňuje připojení ke konzoli Asterisku a ovládat ji přes TCP/IP protokol. Konfigurace AMI se provádí editací konfiguračním souboru, konkrétně souboru `manager.conf`. V konfiguračním souboru se mimo jiné nastavují akce, které uživatel, který se připojuje přes AMI ke konzoli smí provádět.

Skripty, které užíváme pro ovládání konzole pomocí AMI volají interní příkazy Asterisku.

Rozlišujeme dva druhy AMI: správce událostí a správce akcí.

#### AMI správce událostí

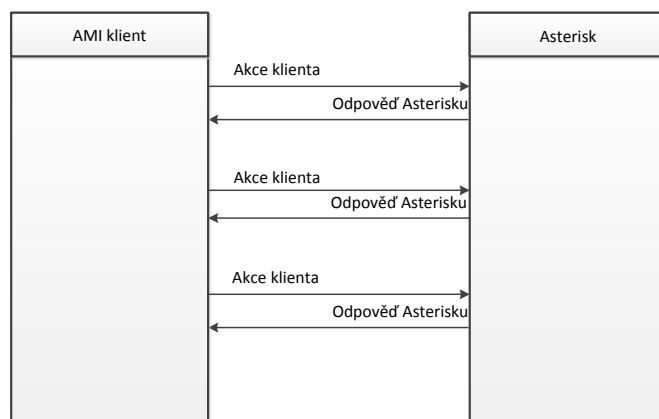
Zobrazuje pouze zprávy směřující z Asterisku k AMI klientovi, tyto zprávy zobrazují stav systému. Události jsou vyvolány samotným Asteriskem.



Obr. 1.2: Komunikace AMI správce událostí

## AMI správce akcí

Správce akcí oproti správci událostí dokáže vyvolávat události (například aktivace telefonního hovoru, ukončení telefonního hovoru, znovunačtení konfiguračních souborů...), komunikace mezi AMI klientem a Asteriskem je tedy obousměrná.



Obr. 1.3: Komunikace AMI správce akcí

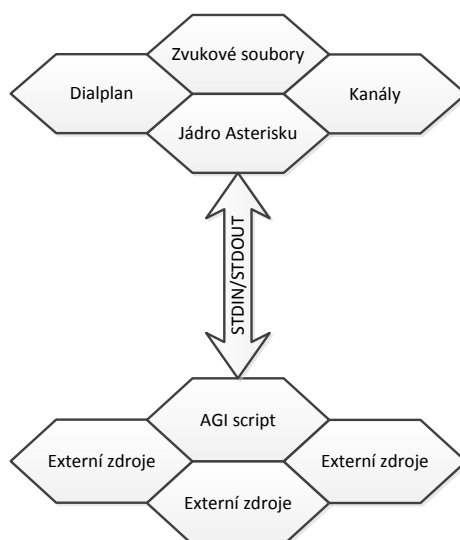
### 1.4.4 Asterisk AGI

Asterisk Gateway Interface umožňuje ovládat ústřednu Asterisk libovolným programovacím či skriptovacím jazykem, nemusíme se tedy omezovat na příkazy konzole Asterisku. S pomocí AGI můžeme kontrolovat například číslovací plán, který se standardně nachází v souboru `extensions.conf`.

Interakce AGI skriptu a ústředny probíhá pomocí standardního vstupu a výstupu STDIN/STDOUT. Komunikace skriptu s Asteriskem je znázorněna na obr. 1.4.

Volání AGI scriptu z dialplanu může vypadat následovně:

```
exten => _X.,1,AGI(jméno_agi_scriptu.agi, parametr1,parametr2...) .
```



Obr. 1.4: Komunikace AGI s Asteriskem

## **2 NÁVRH APLIKACE**

### **2.1 Volba programovacího jazyka**

Pro realizaci mé aplikace jsem se rozhodl využít kombinaci technologií označovaných zkratkou LAMP, tedy Linux, Apache, MySQL a PHP, dále pak technologii HTML a CSS.

#### **2.1.1 PHP**

PHP (Hypertext Preprocessor) je programovací skriptovací jazyk, který je navržen zejména pro vývoj dynamických webových aplikací, nicméně je možné jej použít také pro vývoj desktopových a konzolových aplikací. Skripty jazyka PHP jsou prováděny na straně serveru, k uživateli se dostane pouze výsledek těchto skriptů, který je v mém případě prezentován pomocí technologií HTML a CSS. Pro chod PHP je potřeba mít nainstalovaný a správně nakonfigurovaný webový server, nejčastěji se volí multiplatformní softwarový server Apache.

#### **2.1.2 PHP a databáze**

PHP umožňuje použití funkcí pro přístup k většině běžně používaných databázových systémů (např. MySQL, PostgreSQL, ODBC, Oracle, MSSQL), v mé aplikaci jsem se rozhodl pro uchovávání některých konfiguračních souborů Asterisku využít databázového systému MySQL.

### **2.2 Propojení aplikace s Asteriskem**

Rozhodl jsem se umožnit komunikaci mé aplikace s PBX Asterisk dvojím způsobem, komunikace pomocí realtime módu Asterisku a komunikace pomocí konfiguračních souborů.

#### **2.2.1 Realtime mód**

Celá komunikace aplikace s ústřednou probíhá pomocí tabulek v MySQL databázi, změna jakékoliv hodnoty v tabulce se na ústředně projeví bez nutnosti znovunačtení konfiguračních souborů.

### 2.2.2 „TXT“ mód

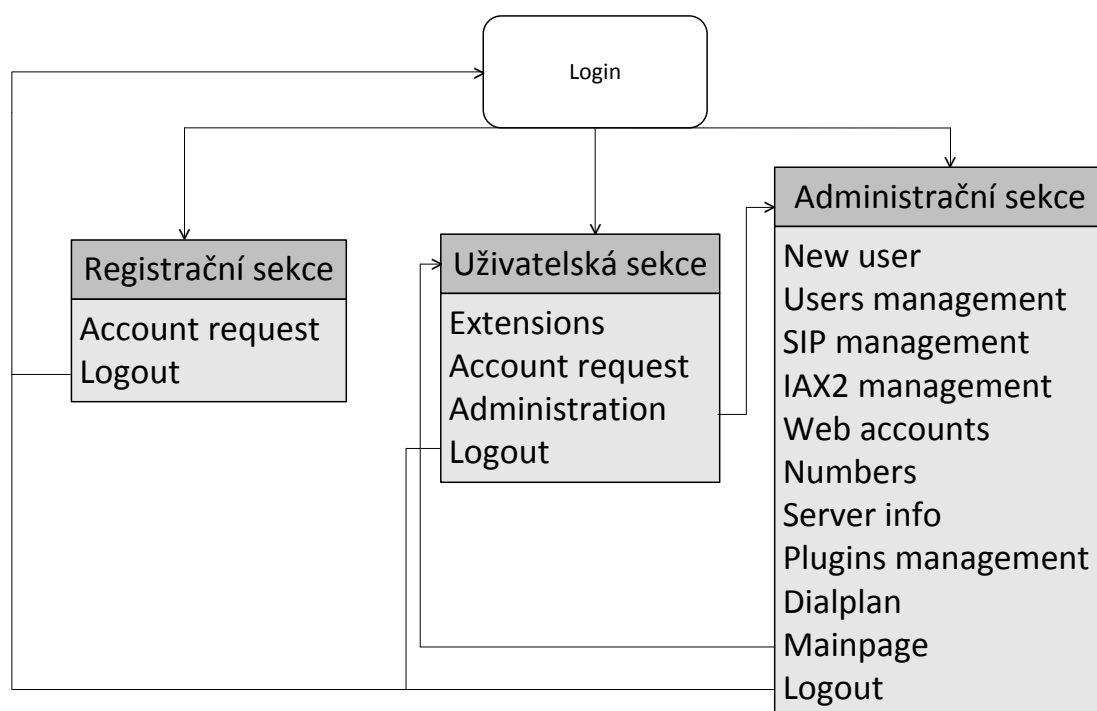
Komunikace s Asteriskem probíhá pomocí editace konfiguračních souborů ústředny, použití toho nastavení aplikace je vhodné tam, kde je nutno šetřit systémové zdroje serveru a nepočítá se s použitím ústředny pro velký počet uživatelů.

V tomto módu je webová část aplikace značně zjednodušená, obsahuje pouze administrační sekci pro základní správu uživatelských účtů.

Protože Asterisk neumí detekovat změnu v konfiguračních souborech, je třeba po každé provedené změně v konfiguračním souboru ústředně říct, že je potřeba znovu načíst konfigurační soubory. K tomuto účelu obsahuje webová aplikace tlačítko, které pomocí Asterisk Manager Interface donutí Asterisk znovu načíst konfigurační soubory.

## 2.3 Členění realtime módu aplikace

Aplikaci jsem se rozhodl členit do tří hlavních sekcí - sekce pro registraci, sekce pro běžné uživatele a sekce pro administrátora. Na obr. 2.1 je znázorněna architektura aplikace.



Obr. 2.1: Architektura aplikace

### 2.3.1 Registrační část

Tato část aplikace je přístupná po přihlášení na účet Guest a slouží k podání žádosti o webový účet. Je nutné vyplnit všechna pole, včetně ověřovacího captcha kódu. Pokud je již uživatelské jméno použito, nebo je špatně opsán captcha kód, uživatel o této skutečnosti bude informován. Podle nastavení v administraci bude účet aktivní automaticky, nebo jej bude muset aktivovat administrátor ručně.

Pro captcha kontrolu jsem použil již existující open source řešení **EasyCaptcha**, šířené pod BSD licenci.

### 2.3.2 Uživatelská část

#### Extensions

V této části uživatel vidí všechny své SIP a IAX účty. U každého účtu je možnost změnit heslo. Po kliknutí na jméno účtu se na konci výpisu účtů zobrazí formulář pro změnu hesla. Odesláním formuláře se změní heslo k danému účtu a uživatel je přesměrován zpět na výpis svých účtů.

#### Account request

Zde si může uživatel zažádat o nový nebo další účet. Uživatel zadá požadované heslo, zvolí požadovaný rozsah čísel, ze kterého se bude číslo generovat a vybere si požadovaný protokol.

Po odeslání formuláře se do příslušné tabulky MySQL databáze vloží řádek, který má vyjma hodnot definujících uživatele shodné hodnoty jako „default“ řádek. Takto vytvořený účet má nastavený kontext na hodnotu register, pokud není v souboru **extensions.conf** kontext register definován, účet nebude funkční. Administrátor může ve zmíněném souboru nastavit chování účtů v kontextu register - například možnost pouze přijímat hovory.

### 2.3.3 Administrační část

Tato část aplikace běžnému uživateli není přístupná, obsahuje následující položky.

#### New user

Zde může administrátor vytvářet uživatelské účty. Nejprve vybere webový účet, ke kterému se má daný SIP či IAX účet přiřadit, poté zadá heslo, vybere sadu čísel pro generování čísla účtu a vybere protokol pro požadovaný účet. Po odeslání se do příslušné tabulky MySQL databáze vloží řádek, který má vyjma hodnot definujících uživatele shodné hodnoty, jako „default“ řádek.

## Users management

Na této stránce může administrátor spravovat veškeré účty Asterisku, které jsou uloženy v MySQL databázi. V uvedené tabulce vidí administrátor jméno vlastníka účtu a jméno účtu. U „čerstvě“ vytvořených účtů je ještě poznámka, že účet má nastaven kontext register a je třeba jej změnit. U ostatních účtů je tlačítko pro smazání účtu. Dále je po kliknutí na jméno účtu možno editovat nastavení tohoto účtu.

## SIP management

Tato stránka umožňuje „přednastavení“ protokolu SIP. Tato možnost je přidána zejména pro ušetření práce administrátora, kdy se nastaví hodnoty pro nově vytvářené účty protokolu SIP. Tato funkce je realizována editací „default“ řádku tabulky sippeers v databázi. Při vytváření nových účtů se hodnoty pro dané položky načítají z „default“ řádku v databázi.

## IAX2 management

Tato stránka umožňuje „přednastavení“ protokolu IAX2. Funkčně je identická se stránkou SIP management.

## Numbers

V této části aplikace může administrátor spravovat sady telefonních čísel. Při vytváření uživatele se náhodně vygeneruje číslo, které leží ve zvoleném intervalu.

## Server info

V této části aplikace vidí administrátor základní údaje o serveru, na kterém je aplikace instalována.

## Web Accounts

Zde může administrátor spravovat účty pro přihlašování do této webové aplikace. Uživatelský účet je definován uživatelským jménem, dále heslem, které se do databáze ukládá jako SHA256 hash a oprávněním. Na výběr jsou 4 úrovně oprávnění:

- Guest;
- User;
- Operator;
- Administrator.

Uživatelské účty je možno vytvářet, editovat a mazat. Pokud se Účet smaže, smažou se také veškeré SIP a IAX účty daného uživatele.

V poslední řadě je možnost zde nastavit, zda budou žádosti o webové účty, o které bylo zažádáno, aktivní bezprostředně po registraci, nebo je bude nutné aktivovat ručně.

## Dialplan

V části dialplan se do formuláře načte obsah souboru `extensions.conf`, ve kterém je definováno samotné chování Asterisku při zpracování hovorů. Po modifikaci a kliknutí na tlačítko **Save** se provede uložení změn souboru `extensions.conf` a provedení příkazu `reload` na konzoli Asterisku pomocí skriptu využívajícího Asterisk Manager Interface.

## Plugins management

V této části aplikace je možno spravovat přídatné moduly. V uvedené tabulce je vypsán název přídatného modulu, sekce, kde se bude zobrazovat odkaz na spuštění daného modulu (uživatelská sekce/administrační sekce), minimální potřebné oprávnění pro zobrazení odkazu pro přechod na příslušný přídatný modul a odkaz pro smazání „instalovaného“ modulu. Po kliknutí na název modulu se zobrazí názvy všech souborů, které daný modul obsahuje. U každého souboru jsou napsány základní informace, dále je možné soubor stáhnout či smazat. Na stránce se dále nachází tlačítko pro přidání dalších souborů do modulu.

**Přidání modulu** Moduly je možno přidávat kliknutím na tlačítko **new plugin**. Po kliknutí na tlačítko aplikace přesměruje administrátora na stránku s formulářem, do kterého je nutné zadat informace nutné pro přidání a správnou vazbu webové aplikace s přídatným modulem. Do formuláře je nutné zadat jméno, pod kterým se daný modul bude zobrazovat v aplikaci, jméno adresáře, ve kterém budou uchovávány soubory modulu na serveru dále pak zvolit minimální práva uživatele, který bude vidět odkaz vedoucí ke spuštění přídatného modulu a zvolit sekci, do které bude odkaz umístěn. Po kliknutí na tlačítko **next** je administrátor přesměrován na stránku s formulářem pro nahrání souborů přídatného modulu. Po zvolení souborů modulu a kliknutí na tlačítko **upload** je přidání modulu dokončeno a administrátor je přesunut na stránku s přehledem „instalovaných“ modulů.

**Požadavky na přídatné moduly** Přídatný modul musí obsahovat soubor s názvem `index`, bez něho nebude modul v kombinaci s aplikací řádně fungovat. Pokud modul využívá CSS styly a nejsou začleněné přímo v kódu modulu, musí být přiloženy



soubory těchto stylů. Pokud chce administrátor do menu umístit odkaz například na stránky VUT, vytvoří soubor se jménem `index.php` a vloží do něj následující kód.

```
<?php
header("Location: http://www.vutbr.cz");
?>
```

Dále postupuje jako v případě přidávání běžného modulu.

## 2.4 Členění „TXT“ módu aplikace

Tento mód aplikace byl vytvořen pro nasazení na servery, kde musíme brát ohledy na systémové prostředky a nepočítáme s velkým počtem uživatelů. V tomto módu můžeme vytvářet účty pouze se základním nastavením nutným pro funkčnost účtu.

### 2.4.1 New user

V této části aplikace administrátor vytváří SIP či IAX účty.

### 2.4.2 Users management

Zde jsou v uvedené tabulce vypsány všechny účty, které jsou definované v konfiguračních souborech Asterisku.

Tyto účty je možno editovat kliknutím na jejich název, či je smazat kliknutím na odkaz `delete`.

### 2.4.3 Server info

Tato část je shodná s částí z Realtime módu 2.3.3.

### 2.4.4 Dialplan

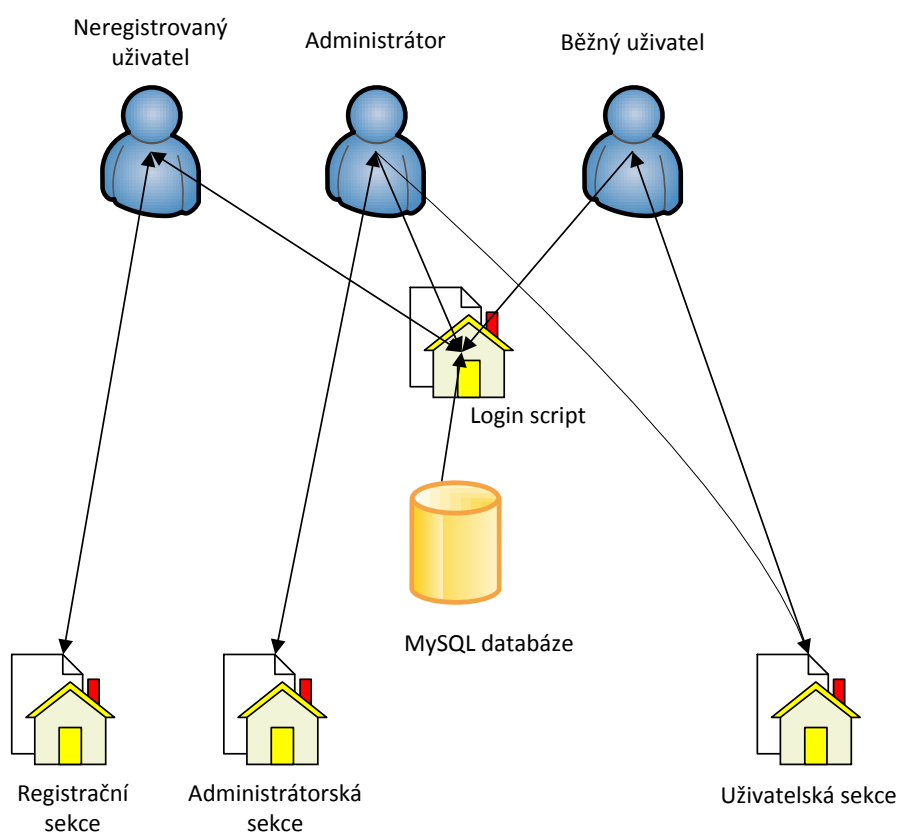
Tato část je shodná s částí z Realtime módu 2.3.3.

### 2.4.5 Reload Asterisk

Kliknutím na odkaz vedoucí na tuto část aplikace se pomocí Asterisk Manager Interface provede na konzoli Asterisku příkaz `reload`, který donutí ústřednu znovu načíst konfigurační soubory. Na stránce se vypíše log z provedeného připojení ke konzoli Asterisku a provedení příkazu `reload`.

## 2.5 Zabezpečení aplikace

Do webové aplikace se uživatel dostane pouze po přihlášení pomocí svého uživatelského účtu. Uživatelský účet je definován přihlašovacím jménem a heslem. Pro zajištění bezpečnosti odesílám otisk hesla získaný pomocí hashovací funkce SHA256, která vrací řetězec fixní délky (256 bitů). Otisk se porovná s otiskem v databázi, který byl získán při žádosti o webový účet. Pokud jsou otisky shodné, aplikace dle oprávnění přeměruje do příslušné části aplikace. Proces přihlašování je znázorněn na obr. 2.2.



Obr. 2.2: Proces přihlašování do aplikace

Po úspěšné autentizaci si klient a server uloží informace o společné relaci. Má aplikace v této relaci uchovává proměnné **username** - definuje jméno uživatele **logged** - říká, zdali je uživatel přihlášen a **permission** - definuje přístupová práva uživatele. Veškeré skripty tyto proměnné kontrolují, dle jejich obsahu určují zda vůbec, případně kam může uživatel vstoupit. Při procesu odhlašování skript tato

data zničí a přesměruje uživatele na stránku pro přihlášení. Odhlášení z aplikace je možné třemi způsoby:

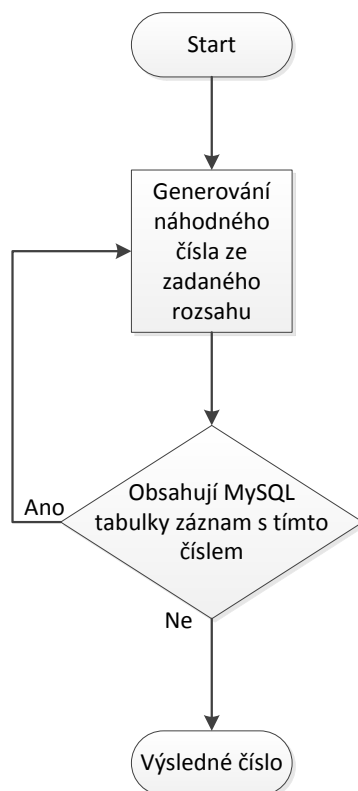
- odhlášením se z aplikace pomocí odkazu vedoucího k odhlašovacímu skriptu;
- nečinností určitou dobu - *tato doba je definována v souboru `php.ini`, který je umístěn na serveru v adresáři `/etc`;*
- zavřením okna prohlížeče.

## 2.6 Generování čísla účtu

Pro generování telefonního čísla jsem vytvořil rekurzivní funkci, která má jako vstupní argumenty rozsah generovaných čísel a údaje pro provedení SQL dotazu na databázi. Tato funkce nejprve vygeneruje náhodné číslo ze zadaného intervalu, které se následně pokusí najít v tabulkách uživatelských účtů v databázi. Pokud skript v databázi nenalezne záznam s tímto číslem, vygenerované číslo je volné a může se přidělit uživateli. Pokud ovšem vygenerované číslo v databázi již existuje, funkce zavolá sama sebe se stejnými argumenty a celý proces kontroly se opakuje. Vývojový diagram této funkce je znázorněn na obr. 2.3. Ve skriptu php vypadá kód takto:

```
function rekurze($from, $to, $prijem)
{
    $extension = rand($from, $to);
    $dotaz = "SELECT * FROM sippeers where name = $extension";
    $dotaz2 = "SELECT * FROM iaxfriends where name = $extension";
    $vysledek = mysql_query($dotaz, $prijem);
    $vysledek2 = mysql_query($dotaz2, $prijem);
    if(mysql_num_rows($vysledek) == 0 &&
        mysql_num_rows($vysledek2) == 0){
        return $extension;
    }else rekurze($from, $to, $prijem);
}
```

Pro generování náhodných čísel jsem se rozhodl z důvodu menší časové náročnosti oproti přidělování čísel „popořadě“ - při velkém počtu již existujících účtů by trvalo nalezení nejvyššího volného čísla déle, než přiřazování náhodně s následnou kontrolou existence čísla.



Obr. 2.3: Generování čísla účtu

## 2.7 Billing modul

Pro demonstrování možností přídatných modulů jsem vytvořil jednoduchý přídatný modul pro správu účtování. Tento modul využívá možnosti Asterisku fungujícího v realtime módu ukládat takzvané call data records - CDR do MySQL databáze.

Modul má dvě sekce, sekci pro běžné uživatele a sekci pro administrátora.

### 2.7.1 Uživatelská sekce

V této sekci jsou v tabulce vypsané proběhlé hovory daného uživatele. Ve výpisu jsou zahrnuty následující informace:

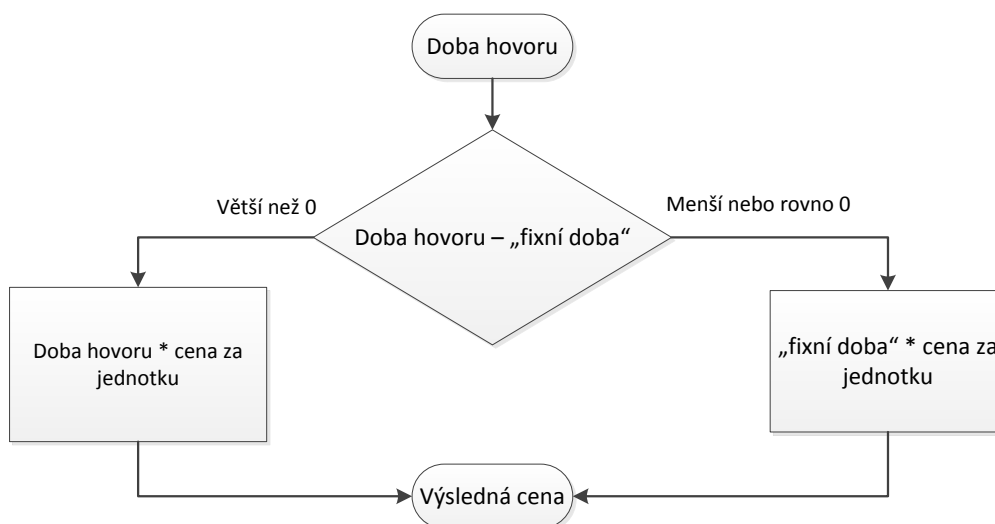
- datum a čas hovoru;
- číslo volaného účtu;
- číslo volajícího účtu;
- celkovou dobu hovoru;
- cenu za hovor.

Na posledním řádku tabulky je vypsána celková doba a cena provedených hovorů.

### 2.7.2 Administrační sekce

Zde může administrátor spravovat účtování hovorů pro jednotlivé účty. V uvedené tabulce je vypsán seznam uživatelů, jejich telefonní účty dále cena za jednotku a „fixní cena“. Pokud chceme v případě nastavování ceny za jednotku použít desetinné číslo, je nutné místo desetinné čárky použít tečku.

Vhodnou kombinací „fixní ceny“ a ceny za jednotku se dá nastavit pevné účtování například za první minutu a následně po první minutě účtování po vteřině. Výpočet výsledné ceny hovoru je znázorněn na obr. 2.4.



Obr. 2.4: Výpočet ceny hovoru

## 2.8 Příklad procesu žádosti o účet

### 2.8.1 Žádost o webový účet

Uživatel po přihlášení na účet **guest** s heslem **guest** zadá v příslušném formuláři své jméno, příjmení, požadované heslo k webovému účtu, vybrané login jméno, svou adresu a opíše ověřovací captcha kód. Po odeslání formuláře skript vyhodnotí, zda byl správně opsán captcha kód a zkontroluje, zdali neexistuje jiný uživatel se zadaným login jménem. Pokud kontrola neproběhla úspěšně, je uživatel vyzván ke opětovnému zadání příslušné informace, v opačném případě je uživatel **guest** odhlášen z aplikace.

Nyní, pokud je v administraci nastavena automatická aktivace webových účtů je účet aktivní, v opačném případě se uživatel sice přihlásí, ale nemůže si podat žádost o telefonní účet.

Aktivaci účtu provede administrátor v administrační části aplikace, v sekci **Web accounts**. U účtů které jsou neaktivní je v položce oprávnění hodnota **inactive**, vedle oprávnění je odkaz **Activate**, který danému účtu nastaví práva běžného uživatele. Pro nastavení vyšších práv je nutné kliknout na jméno účtu a ruční vybrání požadovaného oprávnění s následným kliknutím na tlačítko submit.

### 2.8.2 Žádost o telefonní účet

Po přihlášení uživatele na svůj účet je uživatel přesměrován do uživatelské sekce aplikace. Zde si kliknutím na Account request může podat žádost o telefonní účet. Uživatel je vyzván k zadání hesla, opětovné zadání hesla pro možnost detekce překlepu, dále si vybere sadu čísel, ze které se bude jeho číslo generovat a protokol, jež hodlá využívat. Po odeslání formuláře je uživatel přesměrován na seznam telefonních účtů, vztahujících se k jeho webovému účtu.

Telefonní účet vyžaduje aktivaci od administrátora, chování neaktivovaného účtu definuje kontext **register** v dialplanu Asterisku.

Skript, který vytváří telefonní účet vypadá následovně:

```
if(($password && $password2 && $protocol) == "1") and
($password == $password2)) {
    switch ($protocol)
    {
        case "IAX2":
            mysql_query("INSERT INTO iaxfriends (id, name,...
            SELECT $idcko, $ex, 'type'...
            FROM iaxfriends WHERE fullname = 'default'");
            mysql_query("UPDATE iaxfriends SET secret='$password',
```

```

        fullname='$username', context='register', defaultuser='$ex',
        accountcode='$username' WHERE name='$ex'");
        header("Location: $server/user.php?p=extensions");
break;
case "SIP":
    mysql_query("INSERT INTO sippeers (id, name,...
    SELECT $idcko, $ex, 'type'...
    FROM sippeers WHERE fullname = 'default'");
    mysql_query("UPDATE sippeers SET secret='$password',
    fullname='$username', context='register', defaultuser='$ex',
    accountcode='$username' WHERE name='$ex'");
        header("Location: $server/user.php?p=extensions");
break;
};
} else{
echo "something is wrong";
}; .

```

Skript nejprve zkontroluje úplnost údajů a zkontroluje shodu hesla. Tyto údaje skript získá odesláním formuláře s žádostí metodou **POST**. Na základě uživatelem zvoleného protokolu se následně v databázi vytvoří záznam, který bude mít hodnoty shodné s „default“ záznamem. Následně se vytvořenému záznamu nastaví zadané hodnoty které definují uživatele. Pokud nejsou některé údaje ve formuláři vyplněné, nebo se neshodují hesla, skript nic do databáze nevloží a vypíše že je něco špatně.

Nyní je potřeba takto vytvořený účet aktivovat. Aktivaci účtu provádí administrátor, který v sekci **Users management** vidí u účtů, které je potřeba aktivovat poznámku **Activate**. Skript provádějící výpis uživatelů kontroluje jejich kontext, pokud má kontext hodnotu **register**, přidá k účtu odkaz **Activate**. Odkaz **Activate** vede na stránku, kde může Administrátor provést kompletní nastavení uživatelského účtu. Po odeslání formuláře s nastavením uživatelského účtu skript pomocí metody **GET** načte z URL číslo telefonního účtu a následně vyhledá řádek v tabulce databáze, u kterého je položka **name** shodná se získaným číslem účtu. Skript následně provede SQL dotaz **UPDATE** na vyhledaný řádek, čímž se editují zvolené vlastnosti účtu.

## 2.9 Instalace aplikace

### 2.9.1 Požadavky aplikace na sw vybavení serveru

#### System

Aplikace byla vyvíjena na serveru s operačním systémem Ubuntu verze 12.04.2.2 LTS a počítá s operačními systémy platformy Linux.

#### Apache

Pro správnou funkci webové aplikace je potřeba mít nainstalován Apache verze 2.2.22 a vyšší.

#### MySQL

Databázový systém MySQL verze 5.5.29 a vyšší s vytvořeným účtem pro přístup aplikace k databázi.

#### Asterisk

Aplikace počítá s Asteriskem verze 11.2.1 LTS, která má dlouhodobou podporu. Pro použití s nižší verzí Asterisku je třeba pozměnit několik názvů sloupců v MySQL tabulkách.

### 2.9.2 Import tabulek MySQL databáze

Tabulky pro Asterisk nemusíme vytvářet ručně, tabulky s odpovídající strukturou obsahují zdrojové soubory mé aplikace, stačí je v aplikaci PHPmyAdmin pouze importovat do vytvořené databáze asterisk. Tabulky pro import nalezneme v adresáři /MYSQL. Pokud bychom chtěli aplikaci rozšiřovat o některý přídatný modul využívající RealTime mód Asterisku, který v mé aplikaci nepoužívám, struktury tabulek pro použití MySQL databáze se nacházejí v adresáři /contrib/realtime/mysql instalačního balíčku Asterisku. Mnou použité tabulky jsou vyobrazeny na obr. B.

### 2.9.3 Konfigurace Asterisku

Pro korektní zobrazování uživatelů, „fungujících“ v RealTime módu, v konzoli Asterisku je nutné editovat soubor `sip.conf` a `iax.conf`. Oba soubory jsou v adresáři `/etc/asterisk/`. Do tohoto souboru je nutné do sekce `[general]` vložit následující řádek:

```
rtcachefriends=yes .
```



## Propojení Asterisku s MySQL databází

Nejprve musíme propojit Asterisk s MySQL databází. Nastavení se provádí v souboru `res_config_mysql.conf`, který se nachází v adresáři `/etc/asterisk/`.

Do souboru vložíme následující údaje:

```
[general]
dbhost = 127.0.0.1
dbname = asterisk
dbuser = root
dbpass = password
dbport = 3306
```

`dbhost` říká, jaká je adresa serveru, na kterém se nachází MySQL databáze. `dbname` definuje název databáze, kterou bude Asterisk používat. `dbuser` a `dbpass` říká, pod jakým uživatelským jménem a heslem se asterisk bude k databázi přihlašovat. `dbport` říká, jaký port má Asterisk použít pro připojení k databázi.

Nyní musíme Asterisku říci, „odkud si má které informace brát“. V adresáři `/etc/asterisk/` nalezneme soubor `extconfig.conf`. V souboru se nachází následující řádky, které definují typ databázového systému a název odpovídající tabulky, v mém případě obsah tohoto souboru vypadá následovně:

```
[settings]
iaxusers => mysql,general,iaxfriends
iaxpeers => mysql,general,iaxfriends
sippeers => mysql,general,sippeers
;extensions => mysql,general,extensions
;meetme => mysql,general,meetme
;voicemail => mysql,general,voicemail_users
;queues => mysql,general,queue_table
;queue_members => mysql,general,queue_member_table.
```

Po provedení těchto kroků bude Asterisk pro načítání a uchovávání informací definujících SIP a IAX účty využívat tabulky MySQL databáze, ke které jsme nastavili přístup v souboru `res_config_mysql.conf`. Pokud bychom chtěli vytvořit přídatný modul využívající RealTime architekturu některé „části“ Asterisku, příslušný řádek editujeme.

## Nastavení AMI

Aby správně fungoval AMI skript mé aplikace, je nutné upravit soubor `manager.conf`. nacházející se v adresáři `/etc/asterisk/`.

Náš soubor bude obsahovat tyto řádky:

```
[general]
enabled = yes
port = 5038
bindaddr = 127.0.0.1
[admin]
secret = pw
read = system,call,log,verbose,command,agent,user
write = system,call,log,verbose,command,agent,user
```

Sekce `[general]` definuje údaje pro připojení ke konzoli Asterisku, `[admin]` poté definuje účet pro připojení včetně výčtu úkonů, které může daný uživatel provádět.

### 2.9.4 Instalace vlastní aplikace

Do adresáře na serveru, určenému pro data WWW stránek zkopírujeme veškeré soubory mé aplikace obsažené ve složce `/web`. Dále je nutné ve složce `/etc/asterisk` vytvořit složku s názvem `txt`. Následně je nutné editovat vybrané soubory aplikace. Nejdůležitější soubor, který musíme editovat je soubor, který definuje připojení aplikace k MySQL databázi a mód aplikace, který chceme využívat. Soubor se jmenuje `config.php` a obsahuje následující řádky:

```
$realtime = "mysql";
$txtlogin="admin";
$txtpass="admin";

$server1 = "localhost:3306";
$login = "root";
$heslo = "asdf";
$db = "asterisk";.
```

Proměnná `$realtime` dle hodnoty (`mysql/txt`) určuje, zdali aplikace bude fungovat v RealTime módu, nebo „TXT“ módu. Pokud aplikace bude fungovat v „TXT“ módu, je nutné nastavit proměnné `$txtlogin` a `$txtpass`, které definují účet a heslo pro vstup do „TXT“ sekce aplikace. Proměnnou `$server1` nastavujeme adresu serveru, na kterém jsou umístěny MySQL tabulky. Pro správnou interakci Asterisku

s mou aplikací je bezpodmínečně nutné, aby nastavená adresa byla stejná jako adresa nastavená v souboru `res_config_mysql.conf`. `$login` a `$heslo` říká s jakými přihlašovacími údaji se aplikace bude přihlašovat k MySQL databázi. `$db` definuje jméno databáze, ve které jsou uloženy jednotlivé tabulky.

Dále je nutné editovat soubor `connectdb.php`. V souboru najdeme proměnnou `$server` a nastavíme jí požadovanou hodnotu. Tato proměnná definuje adresu serveru, na kterém je aplikace instalována. V mém případě je tato proměnná nastavena na hodnotu `"http://192.168.10.82/janda"`.

V poslední řadě je nutné do souboru `sip.conf`, který se nachází v adresáři `/etc/asterisk` vložit do sekce `[general]` řádek `#include ./txt/sip_txt.conf` a do souboru `iax.conf`, který se nachází v téže adresáři vložit do sekce `[general]` řádek `#include ./txt/iax_txt.conf`.

## 2.9.5 Nastavení přístupových práv

Pro správné fungování aplikace je důležité nastavit oprávnění k následujícím souborům a složkám:

- soubor `extensions.conf`, nacházející se ve složce `/etc/asterisk`
- složka `txt`, nacházející se ve složce `/etc/asterisk`
- soubor `plugins.TXT`, nacházející se ve složce s aplikací
- složka `plugins`, nacházející se ve složce s aplikací.

Přístupová práva pro soubory nastavíme příkazem: `chmod 1777 jméno_souboru` ze složky se souborem. Změnu přístupových práv pro adresáře provedeme příkazem `chmod 1777 cela_cesta_k_adresari`, příkaz pro změnu práv ke složce `txt` bude tedy vypadat takto: `chmod 1777 /etc/asterisk/txt`.

Pro provádění veškerého nastavení je nutné mít práva superuživatele.

### 3 ZÁVĚR

Cílem práce bylo navrhnout a vytvořit webové rozhraní, pro správu a konfiguraci softwarové ústředny Asterisk. Vytvořenou aplikaci jsem zabezpečil proti vniknutí neautorizované osoby, rozdělil jsem ji na registrační, uživatelskou a administrační část. Komunikace s ústřednou probíhá dle výběru módu aplikace prostřednictvím SQL tabulek uložených v MySQL databázi, nebo prostřednictvím konfiguračních souborů Asterisku. Aplikace je dostupná na adrese `http://192.168.10.82/janda` přihlašovací jméno pro účet administrátora je `admin`, heslo `root`.

Do aplikace jsem vytvořil rozhraní pro přidávání takřka libovolných přídatných modulů, požadavky na tyto moduly jsou popsány v podkapitole 2.3.3.

Funkčnost aplikace byla otestována softwarovým telefonem Zoiper a X-Lite.

## LITERATURA

- [1] ČÍKA, P. *Multimediální služby..* Vysoké učení technické v Brně, 2012, s. 1-127, ISBN 978-80-214-4443-0.
- [2] LUDÍK, M. *Asterisk Manager Interface AMI* [online]. 2012, [cit. 10. 12. 2012]. Dostupné z URL: <<http://bobanl.blogspot.cz/2012/07/asterisk-manager-interface-ami.html>>.
- [3] LUDÍK, M. *Asterisk Gateway Interface AGI* [online]. 2012, [cit. 10. 12. 2012]. Dostupné z URL: <<http://bobanl.blogspot.cz/2012/07/asterisk-gateway-interface-agi.html>>.
- [4] MEGGELEN, J.V, SMITH, J., MADSEN, L. *Asterisk<sup>TM</sup> The Future of Telephony* [online]. 2011, [cit. 11. 12. 2012]. Dostupné z URL: <[http://www.asteriskdocs.org/en/3rd\\_Edition/asterisk-book-html/asterisk-book.html](http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html/asterisk-book.html)>.
- [5] NOVOTNÝ, V., KOMOSNÝ, D. *Doporučení H.323* [online]. 2002, [cit. 10. 12. 2012]. Dostupné z URL: <<http://www.elektrorevue.cz/clanky/02033/index.html>>.
- [6] WIJA, T., ZUKAL, D., VOZŇÁK, M. *Asterisk a jeho použití* [online]. 2005, [cit. 11. 12. 2012]. Dostupné z URL: <[http://www.cesnet.cz/akce/20051115/pr/voz05\\_asterisk.pdf](http://www.cesnet.cz/akce/20051115/pr/voz05_asterisk.pdf)>.

# SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A/D	analog/digital – analogově/digitální
AGI	Asterisk gateway interface
AMI	Asterisk manager interface
CPU	Central Processing Unit – procesor
HTML	HyperText Markup Language – značkovací jazyk pro hypertext
IAX	Inter-Asterisk eXchange – vlastní protokol Asterisku
IVR	Interactive voice response – interaktivní hlasová odpověď
LTS	Long term support – dlouhodobá podpora
MC	Multipoint Controller – vícebodové řídicí jednotky
MD5	Message-Digest algorithm – algoritmus pro vytvoření jedinečného otisku zprávy
PBX	Private branch exchange – pobočková ústředna
RSA	šifra s veřejným klíčem
SHA	Secure Hash Algorithm – hashovací funkce
SIP	Session Initiation Protocol – protokol pro inicializaci relací
TDM	Time Division Multiplex – časový multiplex
URI	Uniform Resource Identifier – textový řetězec pro přesnou specifikaci zdroje
URL	Uniform Resource Locator – jednotný lokátor zdrojů
VoIP	Voice over IP
STDIN	Standardní vstup
STDOUT	Standardní výstup
PHP	Hypertext Preprocessor
SQL	Structured Query Language
CSS	Cascading Style Sheets – kaskádové styly

## SEZNAM PŘÍLOH

A Příloha	39
B Příloha	42

## A PŘÍLOHA

Asterisk TXT

Logout

Administration

New user

Users management

Server info

Dialplan

Reload Asterisk

Users management

IAX2 extension	
2	delete

SIP extension	
1	delete

Host: dynamic

Password: •

Type: Friend

Context: default

Submit

Obr. A.1: Snímek ze správy uživatelů v TXT módu aplikace







## B PŘÍLOHA

sippeers		
id	integer(11)	
name	varchar(10)	
ipaddr	varchar(45)	
port	integer(5)	
regseconds	integer(11)	
defaultuser	varchar(10)	
fullcontact	varchar(35)	
regserver	varchar(20)	
useragent	varchar(20)	
lastms	integer(11)	
host	varchar(40)	
type	varchar(6)	
context	varchar(40)	
permit	varchar(40)	
deny	varchar(40)	
secret	varchar(40)	
md5secret	varchar(40)	
remotesecret	varchar(40)	
transport	varchar(3)	
dtmfmode	varchar(9)	
directmedia	varchar(6)	
nat	varchar(5)	
callgroup	varchar(40)	
pickupgroup	varchar(40)	
language	varchar(40)	
allow	varchar(40)	
disallow	varchar(40)	
insecure	varchar(40)	
trustpid	varchar(3)	
progressinband	varchar(5)	
promiscdir	varchar(3)	
usedientcode	varchar(3)	
accountcode	varchar(40)	
setvar	varchar(40)	
callerid	varchar(40)	
amaflags	varchar(40)	
callcounter	varchar(3)	
busylevel	integer(11)	
allowoverlap	varchar(3)	
allowssubscribe	varchar(3)	
videosupport	varchar(3)	
maxcallbitrate	integer(11)	
rfc2833compensate	varchar(3)	
mailbox	varchar(40)	
session-timers	varchar(9)	
session-expires	integer(11)	
session-minse	integer(11)	
session-refresher	varchar(3)	
t38pt_usertpsource	varchar(40)	
regexten	varchar(40)	
fromdomain	varchar(40)	

iaxfriends	
id	integer(11)
name	varchar(40)
type	varchar(10)
username	varchar(40)
mailbox	varchar(40)
secret	varchar(40)
dbsecret	varchar(40)
context	varchar(40)
regcontext	varchar(40)
host	varchar(40)
ipaddr	varchar(40)
port	integer(5)
defaultip	varchar(20)
sourceaddress	varchar(20)
mask	varchar(20)
regexten	varchar(40)
regseconds	integer(11)
accountcode	varchar(20)
mohinterpret	varchar(20)
mohsuggest	varchar(20)
inkeys	varchar(40)
outkey	varchar(40)
language	varchar(10)
callerid	varchar(100)
cid_number	varchar(40)
sendani	varchar(10)
fullname	varchar(40)
trunk	varchar(3)
auth	varchar(20)
maxauthreq	varchar(5)
requirecalltoken	varchar(4)
encryption	varchar(20)
transfer	varchar(10)
jitterbuffer	varchar(3)
forcejitterbuffer	varchar(3)
allow	varchar(200)
codecpriority	varchar(40)
qualify	varchar(10)
qualifysmoothing	varchar(10)
qualifyfreqok	varchar(10)
qualifyfreqnotok	varchar(10)
timezone	varchar(20)
adsi	varchar(10)
amaflags	varchar(20)
setvar	varchar(200)
iaxfriendname	varchar(40)

billing	
unit	float
fixed	float
user	varchar(20)

service	
id	integer(3)
from	integer(9)
to	integer(9)

settings	
activation	varchar(4)

cdr	
calldate	integer
clid	varchar(80)
src	varchar(80)
dst	varchar(80)
dcontext	varchar(80)
channel	varchar(80)
dstchannel	varchar(80)
lastapp	varchar(80)
lastdata	varchar(80)
duration	integer(11)
billsec	integer(11)
disposition	varchar(45)
amaflags	integer(11)
accountcode	varchar(20)
userfield	varchar(255)

users	
id	integer(11)
username	varchar(40)
pw	varchar(64)
permission	integer(11)
address	varchar(400)
name	varchar(15)
sumame	varchar(15)

Obr. B.1: Tabulky MySQL databáze